



bug_workflow_attributes

Атрибуты отчёта о дефекте

В зависимости от инструментального средства управления отчётами о дефектах внешний вид их записи может немного отличаться, могут быть добавлены или убраны отдельные поля, но концепция остаётся неизменной.

- **Идентификатор (identifier)** представляет собой уникальное значение, позволяющее однозначно отличить один отчёт о дефекте от другого и используемое во всевозможных ссылках. В общем случае идентификатор отчёта о дефекте может представлять собой просто уникальный номер, но (если позволяет инструментальное средство управления отчётами) может быть и куда сложнее: включать префиксы, суффиксы и иные осмысленные компоненты, позволяющие быстро определить суть дефекта и часть приложения (или требований), к которой он относится.
- **Краткое описание (summary)** должно в предельно лаконичной форме давать исчерпывающий ответ на вопросы «Что произошло?», «Где это произошло?», «При каких условиях это произошло?». Например: «Отсутствует логотип на странице приветствия, если пользователь является администратором». Что произошло? Отсутствует логотип. Где это произошло? На странице приветствия. При каких условиях это произошло? Если пользователь является администратором.

- **Подробное описание (description)** представляет в развёрнутом виде необходимую информацию о дефекте, а также (обязательно!) описание фактического результата, ожидаемого результата и ссылку на требование (если это возможно). В отличие от краткого описания, которое, как правило, является одним предложением, здесь можно и нужно давать подробную информацию. Если одна и та же проблема (вызванная одним источником) проявляется в нескольких местах приложения, можно в подробном описании перечислить эти места.
- **Шаги по воспроизведению (steps to reproduce, STR)** описывают действия, которые необходимо выполнить для воспроизведения дефекта. Это поле похоже на шаги тест-кейса, за исключением одного важного отличия: здесь действия прописываются максимально подробно, с указанием конкретных вводимых значений и самых мелких деталей, т.к. отсутствие этой информации в сложных случаях может привести к невозможности воспроизведения дефекта.
- **Воспроизводимость (reproducibility)** показывает, при каждом ли прохождении по шагам воспроизведения дефекта удаётся вызвать его проявление. Это поле принимает всего два значения: всегда (always) или иногда (sometimes). Можно сказать, что воспроизводимость «иногда» означает, что тестировщик не нашёл настоящую причину возникновения дефекта. Тестировщику нужно потратить много времени на то, чтобы удостовериться в наличии дефекта (т.к. однократный сбой в работе приложения мог быть вызван огромным количеством посторонних причин). Разработчику тоже нужно потратить время, чтобы добиться проявления дефекта и убедиться в его наличии. После внесения исправлений в приложение разработчик фактически должен полагаться только на свой профессионализм, т.к. даже многократное прохождение по шагам воспроизведения в таком случае не гарантирует, что дефект был исправлен (возможно, через ещё 10–20 повторений он бы проявился).
- **Важность (severity)** показывает степень ущерба, который наносится проекту существованием дефекта. В общем случае выделяют следующие градации важности:
 - критическая (critical) — существование дефекта приводит к масштабным последствиям катастрофического характера: потеря данных, раскрытие конфиденциальной информации, нарушение ключевой функциональности приложения и т.д.

- высокая (major) — существование дефекта приносит ощутимые неудобства многим пользователям в рамках их типичной деятельности: недоступность вставки из буфера обмена, неработоспособность общепринятых клавиатурных комбинаций, необходимость перезапуска приложения при выполнении типичных сценариев работы.
 - средняя (medium) — существование дефекта слабо влияет на типичные сценарии работы пользователей, и/или существует обходной путь достижения цели, например: диалоговое окно не закрывается автоматически после нажатия кнопок «ОК»/«Cancel», при распечатке нескольких документов подряд не сохраняется значение поля «Двусторонняя печать», перепутаны направления сортировок по некоему полю таблицы.
 - низкая (minor) — существование дефекта редко обнаруживается незначительным процентом пользователей и почти не влияет на их работу, например: опечатка в глубоко вложенном пункте меню настроек, некое окно сразу при отображении расположено неудобно (нужно перетянуть его в удобное место), неточно отображается время до завершения операции копирования файлов.
- **Срочность (priority)** показывает, как быстро дефект должен быть устранён. В общем случае выделяют следующие градации срочности:
 - наивысшая (ASAP, as soon as possible) срочность указывает на необходимость устранить дефект настолько быстро, насколько это возможно. В зависимости от контекста «насколько быстро, насколько возможно» может варьироваться от «в ближайшем билде» до единиц минут.
 - высокая (high) срочность означает, что дефект следует исправить вне очереди, т.к. его существование или уже объективно мешает работе, или начнёт создавать такие помехи в самом ближайшем будущем.
 - обычная (normal) срочность означает, что дефект следует исправить в порядке общей очерёдности. Такое значение срочности получает большинство дефектов.
 - низкая (low) срочность означает, что, в обозримом будущем, исправление данного дефекта не окажет существенного влияния на повышение качества продукта.

- **Фактический результат (actual result)** - результат, полученный после прохождения шагов к воспроизведению.
- **Ожидаемый результат (expected result)** - описывает ожидаемое поведение ПО после прохождения шагов к воспроизведению.
- **Симптом (symptom)** — позволяет классифицировать дефекты по их типичному проявлению. Не существует никакого общепринятого списка симптомов. Более того, далеко не в каждом инструментальном средстве управления отчётами о дефектах есть такое поле, а там, где оно есть, его можно настроить. В качестве примера есть следующие значения симптомов дефекта: Косметический дефект (cosmetic flaw), Повреждение/ потеря данных (data corruption/loss), Проблема в документации (documentation issue), Некорректная операция (incorrect operation), Проблема инсталляции (installation problem), Ошибка локализации (localization issue), Нереализованная функциональность (missing feature), Проблема масштабируемости (scalability), Низкая производительность (low performance), Крах системы (system crash), Неожиданное поведение (unexpected behavior), Недружественное поведение (unfriendly behavior), Расхождение с требованиями (variance from specs), Предложение по улучшению (enhancement).
- **Комментарий (comments, additional info)** — может содержать любые полезные для понимания и исправления дефекта данные. Иными словами, сюда можно писать всё то, что нельзя писать в остальные поля.
- **Приложения (attachments)** — представляет собой не столько поле, сколько список прикрепленных к отчёту о дефекте приложений (копий экрана, вызывающих сбой файлов и т.д.)

Критичность бага – это атрибут, который характеризует влияние бага на общую функциональность разрабатываемого ПО.

По критичности баги делят на:

- S1. Блокирующий (Blocker).** Всё тестируемое ПО не может работать без устранения бага. Например, приёмник начинает перезагружаться сразу после включения, мы не сможем больше ничего протестировать из-за этого бага.
- S2. Критический (Critical).** Большая часть ПО не может корректно работать. Например, приёмник не может открывать закодированные каналы. До устранения этого дефекта можно протестировать UI, а также функционал, не связанный с расшифровыванием каналов.

S3. Значительный (Major). Блокирует работу одной из основных логических цепочек ПО. Например, неправильное сообщение об ошибке при отсутствии подписки на пакет оператора.

S4. Незначительный (Minor). Не нарушает основные логические цепочки приложения, с ним можно продолжать работать почти без потери качества. Здесь можно привести неточный перевод с русского на английский в меню приёмника.

S5. Тривиальный (Trivial). Эта степень присваивается, когда баг вообще не влияет на общее качество работы ПО. Например, незначительное пересечение элементов в меню.

Приоритет бага — это то, в каком порядке нужно решать проблемы.

Существует три степени приоритетности:

P1. Высокий приоритет (High). Нужно исправить немедленно, потому что баг является крайне важным для всего релиза. Например, старое сообщение об отсутствии подписки на пакет, хотя обновление текстов являлось целью этого релиза.

P2. Средний приоритет (Medium). Точно нужно будет исправить, баг достаточно важен, но не требует немедленного решения. Например, некорректный перевод в меню приёмника.

P3. Низкий приоритет (Low). Нужно будет исправить, но баг не очень важный и не требует немедленного решения. Например, это могут быть баги в функционале, который уже не используется оператором, но ещё не был удалён из кода.

Status (статус)

В этом поле находится актуальная информация о том, в каком состоянии текущая задача. Содержание этого атрибута может варьироваться в зависимости от баг-трекинговой системы. Вы можете столкнуться со следующими обозначениями:

- **New** – новый баг;
- **Feedback** – требуется обратная связь;
- **Acknowledged** – с содержанием документа ознакомились;
- **Accepted** – ошибка воспроизвелась и была подтверждена;
- **Assigned** – исправление ошибки назначено;

- **Resolved** – изменения были внесены;
- **Closed** – дефект больше не воспроизводится.

Таковыми являются основные элементы баг-репорта, с которыми приходится встречаться чаще всего. Но в отчёте могут содержаться и дополнительные поля:

- **Environment/platform** – среда, платформа или операционная система;
- **Fix version** – этап разработки ПО на момент выявления бага;
- **Assignee** – пользователь, которому предстоит утвердить или исправить дефект;
- **Lable** – категория ошибки (текст, визуальные элементы и прочее).

Правила создания дефекта

1. Для начала нужно убедиться, что найденный баг ещё не был оформлен. Следует провести поиск его в соответствующем проекте по всем подходящим ключевым словам и/или полям. Если баг уже есть, следует обновить его описание.
2. Если баг не найден – нажимаем на кнопку создания бага. Не стоит забывать важное правило: один дефект - один баг в трекере.
3. Далее нужно постараться кратко описать, что не работает - это и будет заголовок баг-репорта.
4. После этого перейти к подробному описанию бага: указать шаги к воспроизведению.
5. Указать ожидаемый результат. Можно добавить ссылку на спецификацию.
6. Указать полученный результат.
7. Указать версию ПО, также указать версию окружения.
8. Если необходимо, приложить соответствующие артефакты: логи, скриншоты, дампы и т.д.